

**I Claim:**

1. A computer implemented method of determining the equivalence of a first and a second set of simultaneous linear algebraic equations, each of said equations being of a form:

$$e_{i1}x_1 + e_{i2}x_2 + e_{i3}x_3 + \dots + e_{in}x_n = b_i$$

wherein  $x_j$  are unknowns,  $e_{ij}$  are coefficients, and  $b_i$  are quantities, said coefficients and quantities being known algebraic expressions, said method comprising the steps of:

iteratively eliminating said unknowns from each of said sets of simultaneous linear algebraic equations until each of said equations are in the form:

$$(l_{ii})_k x_i = (r_i)_k$$

wherein  $l_{ii}$  and  $r_i$  are algebraic expressions, and  $k=\{1;2\}$  indicate one of said sets that said equation is derived from; and

comparing, for each of said unknowns, the products  $(l_{ii})_1 * (r_i)_2$  and  $(l_{ii})_2 * (r_i)_1$ , wherein said first and said second set of simultaneous linear algebraic equations are equivalent if said products match for all said unknowns.

2. The computer implemented method according to claim 1, said method further including the initial steps of:

recasting said algebraic expressions into a form of one or more token pairs arranged sequentially in a string, each said token pair comprising an operator followed by an operand; and

reducing said strings in accordance with a set of predetermined simplifying rules to obtain reduced expressions; and

wherein said eliminating step is performed on said reduced strings in accordance with a set of predetermined operations.

3. The method according to claim 2, wherein said simplifying rules comprise performing the steps of:

arranging token pairs into subgroups;

arranging operand tokens in an arranged subgroup in order;

reducing the ordered operands by consolidating one or more constants and eliminating variables of opposite effect to form reduced subgroups; and

*Sub*  
consolidating one or more multiple instances of similar subgroups, to produce a reduced string.

4. A computational apparatus for determining the equivalence of a first and a second set of simultaneous linear algebraic equations, each of said equations being in the form:

$$e_{i1}x_1 + e_{i2}x_2 + e_{i3}x_3 + \dots + e_{in}x_n = b_i$$

wherein  $x_j$  are unknowns,  $e_{ij}$  are coefficients, and  $b_i$  are quantities, said coefficients and quantities being known algebraic expressions, said apparatus comprising:

- means for iteratively eliminating said unknowns from each of said sets of simultaneous linear algebraic equations until each of said equations are in the form:

$$(l_{ii})_k x_i = (r_i)_k$$

wherein  $l_{ii}$  and  $r_i$  are algebraic expressions, and  $k=\{1;2\}$  indicate one of said sets that said equation is derived from; and

- means for comparing, for each of said unknowns, the products  $(l_{ii})_1 * (r_i)_2$  and  $(l_{ii})_2 * (r_i)_1$ , wherein said first and said second set of simultaneous linear algebraic equations are equivalent if said products match for all said unknowns.

5. The computational apparatus according to claim 4, said apparatus further including:  
means for recasting said algebraic expressions into a form of one or more token pairs arranged sequentially in a string, each said token pair comprising an operator followed by an operand; and

means for reducing said strings in accordance with a set of predetermined simplifying rules to obtain reduced expressions; and

- wherein said means for eliminating operates on said reduced strings in accordance with a set of predetermined operations.

6. The apparatus according to claim 5, wherein said eliminating means performs the predetermined operations of:

- arranging token pairs into subgroups;  
arranging operand tokens in an arranged subgroup in order;  
reducing the ordered operands by consolidating one or more constants and eliminating variables of opposite effect to form reduced subgroups; and

*Sub*  
*ar*  
consolidating one or more multiple instances of similar subgroups, to produce a reduced string.

7. A computer program product carried by a storage medium for determining the equivalence of a first and a second set of simultaneous linear algebraic equations, each of said equations being of a form:

$$e_{i1}x_1 + e_{i2}x_2 + e_{i3}x_3 + \dots + e_{in}x_n = b_i$$

wherein  $x_j$  are unknowns,  $e_{ij}$  are coefficients, and  $b_i$  are quantities, said coefficients and quantities being known algebraic expressions, said computer program product comprising:

a program element for iteratively eliminating said unknowns from each of said sets of simultaneous linear algebraic equations until each of said equations are in the form:

$$(l_{ii})_k x_i = (r_i)_k$$

wherein  $l_{ii}$  and  $r_i$  are algebraic expressions, and  $k=\{1;2\}$  indicate one of said sets that said equation is derived from; and

a program element for comparing, for each of said unknowns, the products  $(l_{ii})_1 * (r_i)_2$  and  $(l_{ii})_2 * (r_i)_1$ , wherein said first and said second set of simultaneous linear algebraic equations are equivalent if said products match for all said unknowns.

8. The computer program of claim 7 further comprising:

a program element for recasting said algebraic expressions into a form of one or more token pairs arranged sequentially in a string, each said token pair comprising an operator followed by an operand; and

a program element for reducing said strings in accordance with a set of predetermined simplifying rules to obtain reduced expressions; and

wherein said program element for eliminating operates on said reduced strings in accordance with a set of predetermined operations.

9. The computer program of claim 8 wherein said program element for eliminating performs the predetermined operations of:

arranging token pairs into subgroups;

arranging operand tokens in an arranged subgroup in order;

*Nov*  
reducing the ordered operands by consolidating one or more constants and eliminating variables of opposite effect to form reduced subgroups; and consolidating one or more multiple instances of similar subgroups, to produce a reduced string.

5  
10. A computer implemented method for determining the equivalence of sets of simultaneous linear algebraic equations (SLAEs), each said set comprising two or more algebraic equations, said method comprising the steps of:

reducing each SLAE to a standard form; and

10 comparing the SLAEs to determine whether equivalence exists.

11. The method of claim 10, wherein said reducing step includes the steps of:

converting each SLAE into a reduced form;

performing an elimination process; and

15 performing a back substitution process generating a two part string array form for each SLAE.

12. The method of claim 10, wherein said comparing step includes the steps of:

forming a product of a part of a string array with a part of another said string array;

20 forming a product of the other part of a string array with the other part of said another string array; and

comparing said respective products for mathematical equivalence.

13. The method of claim 12, wherein, for the case of three or more sets, said comparing step is repeated for combinations of pairs of the total number of sets.

14. A computer implemented method of determining the equivalence of a first and a second set of simultaneous linear algebraic equations (SLAEs), said method comprising the steps of:

30 iteratively eliminating unknowns from each of said sets of SLAEs to place each SLAE in a two-part standard form; and

forming a product of a part of one said standard form equation with a part of another part of another said standard form equation;

*Sub  
at*

~~forming a product of the other part of said standard form equation with the other  
part of said another standard form equation; and  
comparing said respective products for mathematical equivalence.~~

09597478 . 062000